UCRL-AR-212784

A survey of computer codes used in the U.S. Plasma-Facing Component research community[*]

Compiled by
T.D. Rognlien

University of California Lawrence Livermore National Laboratory
Livermore, CA 94551

Specific code information provided by the following authors and users:

J.P. Allain, ANL; D.A. Alman, UIUC; E. Bringa, LLNL; J.N. Brooks, ANL;
T.E. Evans, GA; A. Hassanein, ANL; Z. Insepov, ANL;
G.R. Longhurst, INEEL; N.B. Morley, UCLA; M.-J. Ni, UCLA; M.J. Pattison, MH;
R. Maingi, ORNL; R. Munipalli, HPC; A. Yu. Pigarov, UCSD; T.D. Rognlien, LLNL;
D.N. Ruzic, UIUC; M. Shaheen, UIUC; S. Smolentsev, UCLA; D.P. Stotler, PPPL;
T. Tanaka, SNL; M.A. Ulrickson, SNL; M.V. Umansky, LLNL; J. Veboncoeur, UCB

May 31, 2005

[*] Work was performed under the auspices of US DOE by University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Introduction

This survey of computer codes used within the U.S. DOE Office of Fusion Energy Sciences is the result of a request by Gene Nardella in December, 2004 to the Plasma-Facing Components Steering Committee. Jeff Brooks, Chair of the PFC Steering Committee tasked Tom Rognlien to compile the survey using input from the U.S. PFC community. A questionnaire was developed to obtain the information and to give a common format. It was decided that both codes developed within OFES programs and those obtained from commercial software companies would be included. Numerous researchers responded and initial summaries have been presented to the PFC Steering Committee and at the PFC technical meeting at PPPL on May 9-11, 2005.

This report gives the final compilation of PFC code information. The codes are divided into two main categories: Plasma/Neutral codes and Materials codes. Each of these areas is subdivided in turn to group closely related codes together. In what follows, there are several pages of a summary of the codes reported, with a clear indication of those that are purchased from commercial developers. After the summary, the more detailed questionnaires from each of the codes are given.

PFC Code Survey: Summary

Plasma/neutral codes

Kinetic transport codes

1. REDEP-WBC: 3D,3v Monte Carlo ions, neutrals, Coulomb collisions with background, sputtering and redeposition, Brooks, ANL [J.N. Brooks, Fus. Eng. Design. **60** (2002) 515].
2. BPHI-3D: 3D,3v, steady-state, PIC ions, Boltzmann electrons, sheath formation, Brooks, ANL [J.N. Brooks, D. Naujoks, Phys. Plasmas **7** (2000) 2565].
3. XOOPIC: 2D, 3v, Time-dependent Particle-In-Cell method with Monte Carlo collisions for sheath formation, dynamics; multispecies, being developed for potential use for PFC [Verboncoeur et al., Comp. Phys. Comm. **87** (1995) 199.]
4. MCI: 3D, 2-3v, Monte Carlo ions, neutrals, Coulomb collisions with background, sputtering and transport in full SOL, Evans, GA [T.A. Evans, D.F. Finkenthal et al., J. Nucl. Mater. **266-269** (1999) 1034].
5. DEGAS2: 3D,3v, Monte Carlo neutrals in realistic geometry, Stotler, PPPL [D.P. Stotler et al., J. Nucl. Mater. **290-293** (2001) 967; manual available at http://w3.pppl.gov/degas2].
6. DUSTT: 3D,3v, Monte Carlo dust particles in plasma edge, neutral and ionized, Pigarov, UCSD [S.I. Krasheninnikov et al., Phys. Plasmas **11** (2004) 3141].

Fluid transport codes

7. UEDGE: 2D, plasma/neutral fluids (occasional MC neutrals), impurities, rad. via implicit finite volume, Rognlien, LLNL [T.D. Rognlien, M.E. Rensink, Fusion Eng. Design. **60** (2002) 497; manual available at http://www.mfescience.org].
8. B2.5: 2D, plasma fluids, neutral fluid or MC, impurities, radiation via finite volume, used by Owen, ORNL [B.J. Braams, Contrib. Plasma Phys. **36** (1996) 276].
9. HKH: 1D, SOL plasma properties from fluid model, radiation, from Harrison, Kukushkin, and Hotston; used by Ulrickson, SNL

Integrated packages

10. HEIGHTS: 1D-3D, combines plasma transport, radiation, and PMI in various approximations, Hassanein, ANL [A. Hassanein and I. Konkashbaev, Fus. Eng. Design. **51-52** (2000) 681].

Plasma turbulence codes

11. BOUT: 3D, plasma fluid model in full tokamak geometry, provides plasmas turbulence fluxes to walls, Xu, Umansky, LLNL [X. Xu et al., Phys. Fluids **7** (2000) 1951].

Materials codes

Stress Analysis, Heat Transfer, and CAD Design

12. ABAQUS: 2D and 3D, finite element stress and heat transfer, commercial code used by SNL [www.abaqus.com].
13. CATIA: 3D, CAD package, commercial code used by SNL [http://www.3ds.com/products-solutions/brands/CATIA].
14. PATRAN: 3D?, CAE package for PFC configuration, used as input for ABAQUS analysis, commercial code used by SNL [www.mscsoftware.com/Products].
15. ANSYS and FEMLAB: 1-3D finite element codes for thermal stress/strain, fluid mechanics, and E&M problems, used by UCLA [http://www.ansys.com and http://www.comsol.com].
16. CFD 2000: 2D and 3D, Navier-Stokes fluid eqns., commercial code used by SNL [http:// www.adaptive-research.com].
17. OPERA: 2D, 3D, finite element solution to Maxwell's eqns. in materials, commercial code used by SNL[http://www.vectorfields.com/files/html/products].
18. CUBIT: 3D, mesh generation code for finite elements, SNL [CUBIT, G.D. Sjaardema et al., SAND94-1100 revised, SNL, May 1994.
19. FILM-30: 1D, heat transfer at solid-to-water boundary, SNL [FILM-30, T.D. Marshall, SAND2001-0629, SNL, FEB 2001].
20. PFCHF: 2D, cylindrical, maps B-field lines to surfaces for PFC heat flux, Ulrickson, SNL
21. HEAT-1D: 1D, time, thermal transport in PFC from plasma heat flux, Ulrickson, SNL
22. IHC: 1D, time, material heat conduction with temperature dependent properties, Ulrickson, SNL
23. NdotB: 3D, maps plasma heat flux to PFCs, Ulrickson, SNL
24. TMAP7: 1D, transport of tritium and other gas in materials via diffusion model, Longhurst, INNEL [G.R. Longhurst, TMAP7 Users Manual, INEEL/EXT-04-02352, INNEL, Oct. 2004].

Molecular Dynamics

25. MolDyn: 3D, 3v, Newton eqns for many-body Brenner potentials, Alman, UIUC [D.A. Alman, D.N. Ruzic, J. Nucl. Mater. **313-316** (2003) 182].
26. MDCASK: 3D, 3v, Newton eqns. for many-body Brenner and AIREBO potentials, radiation, surface evolution, Bringa, Gilmer LLNL [http://www.llnl.gov/asci/purple/benchmarks/limited/mdcask].
27. "MD_ANL:" 3D, 3v, Newton eqns. for many-body potentials; have included Li, H, He; Insepov, ANL [Z. Insepov, submitted J. Nucl. Eng. (2005)].

Binary Collision Approximation (BCA)

28. ITMC: 3D, Monte Carlo ion interaction with materials, includes Coulomb collisions, atomic potentials, (BCA model?), Hassanein, ANL [A. Hassanein, et al, Nucl. Instrum. & Meth, **B13** (1986) 25].
29. VFTRIM-3D: 3D, 3v, Monte Carlo particles interaction with materials, BCA model, including rough-surface effects, Shaheen, UIUC [D.N. Ruzic, Nucl. Instr. Methods Phys. Res. **B47** (1990) 118].

Hybrid MD/BCA

30. MD-TRIM-3D; 3D, 3v, combines Monte Carlo and MD many-body effects for sputtering, etc., Allain, ANL, Ruzic, UIUC [J.P. Allain et al., Nucl. Instr. Meth. Phys. Res. B, Submitted 2004].

Liquid flow analysis

31. HIMAG: 1D-3D, Navier-Stokes fluid with MHD via finite difference, liquid flow, Munipalli, HyPerComp [N.B. Morley et al., Fusion Eng. Design **72** (2004) 3].
32. MetaFlow: 1D-3D, fluid equations with MHD via lattice Boltzmann, liquid flow, may be applicable to edge plasmas, Pattison, MetaHeuristics [M.J. Pattison and S. Banerjee NUTHOS-6 Conf., Nara, Japan, 4-8 October 2004].
33. FDMHD: 2D momentum and induction equations for MHD flow that utilizes 3D energy and tritium transport for a multilayer rectangular channel, Smolentsev, UCLA [S. Smolentsev et al., Fusion Eng. Design **73** (2005) 83].
34. TSLAMHD: 3D MHD code using the thin-shear-layer model for free-surface flow in a strong magnetic field, Smolentsev, UCLA [S. Smolentsev et al., Appl. Math. Model. **29** (2005) 215].
35. JHS: 2D and 3D free-surface MHD code for single-phase and two-phase flows. Uses Navier-Stokes equations with level-set method for the surface evolution, Ni, UCLA [M.-J. Ni et al., AIAA Journal **40** (2002) 1464].
36. HEIGHTS: see item 10 above.

PFC Code Survey: Questionnaire Details

The following pages give more detailed code information provided in response to the survey questionnaire. The item number in the code summary above is also used to identify the codes in the remaining section, which are also ordered sequentially.

1. REDEP

Abstract:
REDEP-WBC Code Package, J.N. Brooks author. Computes sputtering and transport of diverter, wall, limiter, etc. surface materials in a tokamak or other plasma device. Computes sputter limited lifetime, impurity contamination of core plasma, and tritium codeposition in redeposited material.

References:
J.N. Brooks, Nucl Tech./Fus. 4(1983)33.
J.N. Brooks, Phys. Fluids 8(1990)1858.
J.N. Brooks, Fus. Eng. Des. 60(2002)515.

Code package is generally used internally at ANL, no user manual

Physical problem being simulated:
Sputtering (physical and chemical) by ions and charge exchange neutrals; transport of impurities in D-T (or other) plasma; tritium codeposition as a function of surface temperature, material, oxygen content; carbon/hydrocarbon chemical sputtering and reactions.

Space and time domain:
3-D in space, 3-D in velocity, time-dependent, arbitrary geometry, time scale = 10^{-9}-10^{-6} s

Underlying physics equations and processes included:
Fully kinetic single particle sub-gyro orbit motion of impurity particle (atom, ion, hydrocarbon molecule/radical), in background hydrogen isotope (or other) plasma. Newton's equation with Lorentz force and collisions. Oblique-incidence sheath parameters. Test particle elastic and inelastic collisions with background plasma. Fokker Plank/extended-Braginski-method (non-disparate mass) collision coefficients + anomalous diffusion. All relevant processes included or includable.

Numerical models:
Monte Carlo

Computer science and numerical algorithms:
FORTRAN

Requirements and performance:
Generally run on single workstation. Memory requirements minor. Some parallelization done/used (ANL IBM-cluster). Typical run = 1 hr.

Verification and validation:

Extensive benchmarking done with plasma/surface interaction data from major fusion devices, e.g., DIII-D, JET, TFTR, and plasma simulators, e.g., PISCES.

Interaction with other codes:
In line coupling with TRIM sputtering. code has been done. Off-line coupling with UEDGE, molecular dynamic codes. It would be highly desirable to directly couple REDEP-WBC with these and related codes.

User community:
3

2. BPHI

Abstract:
BPHI-3D, J.N. Brooks, D. Naujoks authors. Computes self-consistent magnetic and Debye sheath parameters and plasma transport for oblique incidence magnetic field with and without surface emitted/in-sheath-ionized material.

References:
J.N. Brooks, D. Naujoks, Phys. Plasmas 7 (2000) 2565.
D. Naujoks, J.N. Brooks, J. Nucl. Mat. 290-293 (2001) 1123.
(no manual)

Physical problem being simulated:
Oblique incidence, strong magnetic field sheath, for D-T plasma with wall emitted/ionized surface material.

Space and time domain:
3-D, steady state

Underlying physics equations and processes included:
Monte Carlo kinetic treatment for D-T and impurity ions/neutrals
Boltzman/guiding-center electrons
Particle-in-cell time-independent Poisson solver

Numerical models:
Particle-in-cell, Monte Carlo

Computer science and numerical algorithms:
C++
Poison equation solver = implicit alternating direction method for inversion of the tridiagonal matrix.

Requirements and performance:
Generally run on workstation
Running time ~ 2 hrs.

Verification and validation:
Used to explain "superbrilliance event" (runaway carbon sputtering) in TORE SUPRA tokamak.

Interaction with other codes:
Coupled with THERM thermal code, for studying surface temperature limits in liquid lithium surface systems.

User community: ANL and IPP/Berlin personnel.

3. XOOPIC

Abstract:

XOOPIC, J. P. Verboncoeur, A. B. Langdon, N. T. Gladd, many UCB students
2D electrostatic and electromagnetic, Cartesian and axisymmetric, fully relativistic, includes Monte Carlo collision model.
Applications include low temperature discharges (e.g. materials processing), high pressure discharges (e.g. fluorescent lamps, plasma display panels), plasma thrusters, microwave beam devices, satellite charging, beam optics, high energy accelerators (wakefield and conventional), laser plasma interaction, fusion plasmas (sheath formation, lower hybrid heating), many more…

References:

Verboncoeur et al., Comp. Phys. Comm. 87, 199 (1995).
User manual for input parameters only in digital format (in doc/ subdirectory)

Physical problem being simulated:

Sheath formation, chemistry and transport within sheath

Space and time domain:

2D time domain, slab and cylindrical, 3 velocity particle model

Underlying physics equations and processes included:

Newton-Lorentz, Maxwell or Poisson, differential cross section neutral collisions

Numerical models:

Particle-in-Cell
Monte Carlo

Computer science and numerical algorithms:

Programming language: C++
Visualization/graphics: – runtime using Xgraphics
2nd order explicit time advance
2nd order finite difference/finite volume

Requirements and performance:

Platforms: Any unix (there is a commercial version for Windows)
Memory is required: 20M or more, depending on problem size
Typical simulation time: Widely varies: minutes to weeks
Parallel modes: Parallel on Linux clusters, any other MPI platform, up to 64 processors that I know of
Major time-limiting component: For most problems, the particle push and weighting, but for some problems with high vacuum to plasma volume ratio it is the field solve

Verification and validation:
Comparisons with Magic on Child-Langmuir space charge limited flow
Comparisons with ICEPIC on beam and electromagnetic wave problems, multipactor
Comparisons with experiments and theory in many areas

Interaction with other codes:
Coupling to a model like UEDGE would be very valuable, so that fields and particle fluxes could be made consistent

User community:
The entire Berkeley code suite has about 1000 users, with over 150 publications (that I am aware of) in the last 10 years internationally. XOOPIC is the most popular component of the suite.

4. MCI

Abstract:

The Monte Carlo Impurity (MCI) code uses a background plasma (typically from a fluid code such as UEDGE) to study impurity sputtering and transport in DIII-D. The code is quasi-kinetic and follows both impurity neutrals and ions in the full 3D geometry of the tokamak edge, SOL and divertor. It is coupled to both the ADAS and ADPAK atomic database programs and has a variety of physical and chemical sputtering models built into the boundary conditions. It has also been coupled to near wall sputtering/transport solutions from the WBC code.

References:

T. E. Evans, D. F. Finkenthal, M. E. Fenstermacher, A. W. Leonard, G. D. Porter, and W. P. West, "Quantitative Comparisons Between Experimentally Measured 2D Carbon Radiation and Monte Carlo Impurity Code Simulations", *J. Nucl. Mater.* **266-269** (1999) 1034.

T. E. Evans, D. F. Finkenthal, Y. S. Loh, * M. E. Fenstermacher, G. D. Porter, and W. P. West, "Comparisons of physical and chemical sputtering in high density divertor plasmas with the Monte Carlo Impurity (MCI) transport code", *Contr. Plasma Physics,* **38** (1998) 260.

T. E. Evans, and D. F. Finkenthal, "Monte Carlo Impurity Transport Modeling in the DIII-D Tokamak", in preparation for *Atomic Processes in Plasmas*, Ed., E. Oks and M. S. Pindzola, Invited paper, in the American Institute of Physics Conference Proceedings, Vol. **443**, pp 58-72, AIP Press, New York, 1998.

There are no user manuals available at this time, contact T. Evans at GA.

Physical problem being simulated:

Divertor, SOL and edge plasma transport (radiated power and sputtering yields) of intrinsic trace impurities. Non-intrinsic trace impurities may also be introduced as point or distributed sources. Impurities are introduced as cold neutrals and allowed to equilibrate with the background plasma as they are ionized and transported through the divertor, SOL and edge plasma.

Space and time domain:

3D, toroidal (with true flux surface and wall geometry), both time independent and time dependent simulations can be run with time scales from ms to several particle confinement times.

Underlying physics equations and processes included:
Quasi-kinetic (approximate kinetic effects without drifts or velocity space diffusion), uses an approximate collision operator and simulates the effects of ion thermal gradients, sheath potentials, plasma flows and cross field diffusion due to turbulence.

Numerical models:
Monte Carlo

Computer science and numerical algorithms:
FORTRAN with netCDF data files and IDL data visualization tools

Requirements and performance:
What platforms will the code run on? UNIX and LINUX
How much memory is required? 512 Mb
How long does a typical simulation take? 10-15 cpu seconds.
Can the code run in parallel? If so, how many CPUs can be utilized and what platforms have been used in the parallel mode? No (not yet)
What is the major time-limiting component in the code? Transport simulation module

Verification and validation:
Benchmarked against DIVIMP and UEDGE multifluid simulation (with carbon impurities). Attempts to validate physics models (sputtering, radiation, transport, etc.) have not been completely successful to date (see T. E. Evans, D. F. Finkenthal, M. E. Fenstermacher, A. W. Leonard, G. D. Porter, and W. P. West, "Quantitative Comparisons Between Experimentally Measured 2D Carbon Radiation and Monte Carlo Impurity Code Simulations", *J. Nucl. Mater.* **266-269** (1999) 1034 for example).

Interaction with other codes:
UEDGE, b2/b2.5, WBC

User community:
2 at GA in last 3 years (~10 over the last 6-7 years)

5. DEGAS 2

Abstract:
DEGAS 2, written by C. F. F. Karney and D. P. Stotler, is a neutral transport code primarily targeted at simulations of neutral species arising either from plasma-wall interactions or externally controlled gas sources. The code effectively replaces the older DEGAS code, written by D. Heifetz.

References:
D. P. Stotler and C. F. F. Karney, Contrib. Plasma Phys. **34**, 392 (1994).
D. P. Stotler, C. S. Pitcher, C. J. Boswell et al., J. Nucl. Mater. **290-293**, 967 (2001).

User's Manual and other relevant materials available at: http://w3.pppl.gov/degas2

Physical problem being simulated:
Transport of neutral atoms and molecules through plasma and vacuum, including electron impact collisions, neutral-ion and neutral-neutral interactions, and plasma / neutral interactions with material surfaces.

Space and time domain:
The core of DEGAS 2 is inherently 3-D in configuration and velocity space. For simplicity, the most commonly used geometries have one ignorable coordinate. Slab and toroidal configurations can both be handled. Presently, DEGAS 2 is run only in steady-state mode.

Underlying physics equations and processes included:
The fundamental equation is the Boltzmann equation, describing the distribution function of the neutral species. One additional simplification is made in the handling of neutral-neutral interactions. Those collisions are described by a Krook operator; i.e., the collision time is a constant parameter. An iterative solution is required in this case since the equation is rendered nonlinear.

Numerical models:
Monte Carlo

Computer science and numerical algorithms:
DEGAS 2 is written in a "macro-enhanced" version of FORTRAN. The FWEB utility is used to define macros that provide some object oriented capability and deal with tedious tasks such as reading / writing netCDF files and dynamic memory allocation. The FWEB prepocessor can generate from this file either F77 or FORTRAN 90 code suitable for compilation.

DEGAS 2 provides no direct graphics capability. One post-processor generates HDF files that can be viewed with commercially available graphics tools.

Requirements and performance:
DEGAS 2 can be adapted to many UNIX platforms. In the past, it has run on Sun, DEC-Alpha, SGI, Cray, and Linux. Memory requirements vary substantially with problem size, but are kept modest by dynamic memory allocation. The largest problems run to date needed a few hundred megabytes of memory. Likewise, run times can vary substantially. The shortest simulations can take only seconds. The longest, those involving neutral-neutral interactions, can take days or even weeks. For this reason, parallelization was built into the code at the outset. CPU utilization should be efficient as long as the number of CPU's is much less than the number of particles used (typically one million). Parallel jobs have been run on all of the platforms listed above. Some jobs using heterogeneous groups of CPU's were also carried out.

The major time-limiting component in the code is believed to be the evaluation of the function that describes a 3-D quadratic surface.

Verification and validation:
Verification tests are described in the User's Manual. The main benchmarks are against the EIRENE and DEGAS Monte Carlo neutral transport codes. Many less involved tests have not been documented.

Virtually all code-experiment comparison exercises undertaken so far have been hindered by limitations in the available experimental or reference (atomic and plasma-material interaction) data and are not sufficiently rigorous to warrant being labeled as "validation" efforts. These include:
D. P. Stotler, C. H. Skinner, R. V. Budny et al., Phys. Plasmas **3**, 4084 (1996) (original DEGAS code)
D. P. Stotler, A. Y. Pigarov, C. F. F. Karney et al., in **Proceedings of the Sixteenth International Conference on Plasma Physics and Controlled Nuclear Fusion Research**, (Montreal, Canada, September 1996) (International Atomic Energy Agency, Vienna, Austria, 1997), Vol. **2**, p. 633.
D. P. Stotler, C. S. Pitcher, C. J. Boswell et al., J. Nucl. Mater. **290-293**, 967 (2001).
D. P. Stotler, D. A. D'Ippolito, B. LeBlanc et al., Contrib. Plasma Phys. **44**, 294 (2004).

Interaction with other codes:
In terms of input data, DEGAS 2 is coupled to a large number of other codes. For example, virtually all of the atomic and plasma-material interaction data are either generated or manipulated by other codes. These data are put into tabular form for use in DEGAS 2. Also, the most sophisticated DEGAS 2 geometry setup tool utilizes output from the DG and Carre codes. These in turn rely on EQDSK equilibrium files, such as those generated by the EFIT code.

The most relevant coupling is with the UEDGE fluid plasma transport code. DEGAS 2 is able to read the geometry and plasma data generated by UEDGE. In turn, the plasma sources computed by DEGAS 2 can be read by UEDGE, permitting an iterative approach to a self-consistent solution.

User community:
Groups that have used the code recently include: LLNL, JAERI (JT-60U), University of Illinois Urbana-Champaign, KAIST, and KBSI.

6. DUSTT

Abstract:

Authors: A. Yu. Pigarov, T.K. Soboleva, S.I. Krasheninnikov:
The DUST Transport (DUSTT) code simulates the 3D transport of dust particles in plasmas (intrinsic and injected dusts) as well as impurity profiles associated with dust evaporation and related radiation emissivity for dust diagnostics. The code is designed to be coupled to edge-plasma physics code UEDGE in order to study self-consistently the effects of dusts on plasma parameters, plasma contamination by impurities, and erosion/deposition in tokamaks.

References:

This is novel code. The general idea of the code is discussed in [S.I. Krasheninnikov et al "On dust dynamics in tokamak edge plasma", Physics of Plasmas 11 (2004) p.3141].

Physical problem being simulated:

The code incorporates the simple models for generation, acceleration in plasma sheath, transport in edge plasma, collisions with walls and micro-turbulences, surface charging, and ablation of dust particles. The chain of codes: DUSTT + multi-species UEDGE + material-surface evolution code, will be able to solve general plasma/surface problems.

Space and time domain:

The DUSTT code is 3D+3V. It uses 2D plasma profiles from UEDGE assuming toroidal symmetry of a tokamak. As far as the single trajectory simulation is concerned, the code is time-dependent. For coupling with UEDGE, the DUSTT code performs Monte Carlo averaging over an ensemble of test dust particles.

Underlying physics equations and processes included:

The DUSTT code solves 3D+3V equations of motion. It incorporates realistic forces acting on particle in the sheath and edge plasmas and accounts for random collisions with rough material surfaces and plasma micro-turbulence. In addition, the code solves heat transfer and particle flux balance equations for calculating the temporal evolution of temperature, charge, and mass of dust particles.

Numerical models:

The DUSTT code operates on 2D curvilinear non-uniform mesh based on MHD equilibrium and generated by UEDGE. Along with simple solver for a system of differential equations, the code uses Monte Carlo method to perform averaging over initial dust parameters (birth point, velocity vector, mass, radius, and etc) and collisions.

Computer science and numerical algorithms:
Language(s): FORTRAN, C
Visualization/graphics: ascii-file output which can be handled by many visualization tools.

Requirements and performance:
So far, the code is platform independent but Unix environment is preferential. The code can easily be parallel. The major time is obviously spent on 3D trajectory modeling.

Verification and validation:
The code is planned to be validated against experiments on dust imaging with 3 fast cameras that are planned soon on NSTX.

Interaction with other codes:
Meantime, the coupling between DUSTT and UEDGE (and, further, surface evolution code) is by means of the file exchange.

7. UEDGE

Abstract:
UEDGE has many contributors with primary active developers being T.D. Rognlien and M.E. Rensink, LLNL. The code solves multispecies fluid equations for various ion charge states and electrons for strongly magnetized plasmas. The code can use either a fully integrated fluid neutral model or couple to Monte Carlo neutrals. UEDGE is used to simulate 2D plasma profiles, surface heat flux, and impurity transport for tokamaks and other MFE devices. Parallel transport is classical and cross-B-field transport is enhanced to model turbulence.

References:
T.D. Rognlien, M.E. Rensink, Fusion Eng. Design. **60** (2002) 497; Phys. Fluids **9** (2001) 2120. Also, T.D. Rognlien et al., Phys. Fluids **6** (1999) 1851. A user manual is available at http://www.mfescience.org.

Physical problem being simulated:
Fluid transport of plasma ions, electrons, and neutrals for strongly magnetized plasmas. Applications are especially focused on tokamak edge plasmas.

Space and time domain:
2D in space, utilizing slab, cylinder, or toroidal geometry. One spatial dimension is along the magnetic field. The code can simulate accurate time-dependence or perform nonlinear Newton iterations to efficiently find steady-state solutions.

Underlying physics equations and processes included:
Magnetized plasma fluid equations are used including particle continuity, parallel momenta, perpendicular diffusion and convection, ion temperature, electron temperature, and electrostatic potential. Collisional parallel transport is include ala Braginskii together with classical cross-B-field drifts and parameterized turbulence transport. Hydrogen and impurity radiation, ionization, and recombination included via tables (ADAS, STRAHL, or ADPAK).

Numerical models:
The plasma and neutral fluid equations are discretized using a finite-volume method.

Computer science and numerical algorithms:
Mostly FORTRAN (99%), initially F77, and compatibility and some evolution to F90. Remainder is C. Usually run under the BASIS coding shell, but also runs under PYTHON. Stand alone is possible, but not convenient. The time-advance is fully implicit using a finite-difference Jacobian for a preconditioner to allow large time steps. A sparse solver, ILUT, is used to solve the preconditioner and an Newton Krylov algorithm is used both the time-dependent and state-state solutions. NCAR graphics is typically used.

Requirements and performance:
UEDGE runs on many different platforms from workstations (LINUX, SUN, SGI) as well on larger machines (T3E, IBM-SP). Since UEDGE generally runs under BASIS or PYTHON, one of these systems should be present. A typical single ion species simulation requires ~500 MB for a 50x50 mesh, and simulations can take from 10 minutes to several hours. An arbitrary number of impurity species can be included, and the six charge states of carbon, for example, can increase the run time by a factor of 3-6. A parallel version exists under PYTHON, but usually the serial version is used. The computer time is roughly evenly distributed between calculating the preconditioning Jacobian, solving the preconditioner, and solving the full fluid equations.

Verification and validation:
Various verifications have been done with know simple solutions for parallel transport, and with other fluid transport codes such as B2 and BORIS. Numerous validation comparisons have been done with edge data from DIII-D, NSTX, and JT-60U tokamaks, although these use the turbulent radial transport as an adjustable parameter to fit the midplane profiles that then allows comparison of divertor parameters and wall fluxes. Recently, coupling with the BOUT turbulence code has allowed the radial transport to be determined from a fundamental model.

Interaction with other codes:
UEDGE couples to the DEGAS-2 or EIRENE neutral Monte Carlo codes. The code also now couples with the BOUT turbulence code. UEDGE also provides input data for WBC for near-surface plasma and neutral transport, and uses data of impurity sources for possible iterative calculations.

User community:
Within the last several years, UEDGE has been used by the following institutions in the U.S.: LLNL, GA, UCSD, ORNL, PPPL, and U. Texas (IFS). Internationally, it has been used by U. Quebec, JAERI (JT-60U), NIFS (LHD), Korea (KSTAR), and TRINITI.

8. B2.5

Abstract:
b2.5 code, B.J. Braams author. The b2.5 code is the predecessor to UEDGE, and development has continued on b2.5 independently since UEDGE was written. DEGAS is a 3-D Monte Carlo neutral transport code. The code packages were used for 2-D edge plasma simulations of the background plasma in a DIII-D DiMES Lithium exposure experiment, and the plasma solution was given to Todd Evans for MCI Monte Carlo impurity transport calculations.

References:
b2.5:
(1) B.J. Braams, Contrib. Plasma Phys. 36 (1996) 276.
(2) R. Maingi, J.T. Hogan, L.W. Owen, et. al., Nucl. Fusion 34 (1994) 283.
DEGAS
D.B. Heifetz, D. Post, M. Petravic, et. al., J. Comp. Phys. 46 (1982) 309.

No user's manual that I am aware of. Code available from B.J. Braams.

Physical problem being simulated:
2-D Edge plasma transport with a semi-analytic kinetic treatment of the first flight neutrals

Space and time domain:
2-D space, toroidal geometry, steady state

Underlying physics equations and processes included:
Braginskii fluid equations with kinetic corrections for parallel transport; anomalous perpendicular transport.

Numerical models:
Fluid with simple kinetic treatment for first flight neutrals

Computer science and numerical algorithms:
FORTRAN with NCAR graphics and IDL post processor
Finite difference (choice of central or upwind difference) with implicit time advancing

Requirements and performance:
What platforms will the code run on? NERSC, and unix machines
How much memory is required? I do not recall
How long does a typical simulation take? <10 minutes clock time, < 1 min. CPU time
Can the code run in parallel? If so, how many CPUs can be utilized and what platforms have been used in the parallel mode? No
What is the major time-limiting component in the code? Identifying and obtaining data for comparison

Verification and validation:
(1) R. Maingi, Ph.D. thesis, "Coupled 2-D Edge Plasma and Neutral Gas Simulations of Tokamak Scrape-ff Layers", North Carolina State University, 1992.
(2) L.W. Owen, et. al., J. Nucl. Mater. **220-222** (1995) 315-319.
(3) D.R. Baker, et. al., J. Nucl. Mater. **241-243** (1995) 602-605.
(4) L.W. Owen, et. al., J. Nucl. Mater. **266-269** (1999) 890-895.
(5) L.W. Owen, et. al., J. Nucl. Mater. **290-293** (2001) 464-468.

Interaction with other codes:
Done with DEGAS, DEGAS-2, MCI (Evans-GA)

User community:
Maybe 5?

9. HKH SOL Model

Abstract

HKH Edge model, Mike Ulrickson, in MathCAD

References

No publication of code

Physical problem being simulated

Edge plasma transport

Space and time Domain

1D space/ slab/ time independent

Underlying physics equations and processes included

Fluid equations, simplified radiation loss

Numercial Models

Analytic

Computer Science and numerical algorithims

MathCAD Only,

Requirements and performance

PC, 5 s, no limit

Verification and validation

None.

Interaction with other codes

none

User Community

Oak Ridge, Princeton

10. HEIGHTS

Abstract:

HEIGHTS, A. Hassanein and HEIGHTS team

The software is a comprehensive package of several modules (HEIGHTS-IFE, MFE, EUV, MHD, ...), designed to study phenomena such as shock and ignition physics; heat and radiation propagation through different media; plasma dynamics; laser-, ion-, and plasma interaction with targets; thermal response and erosion rates, as well as the lifetime of the structural components, being deposited by particles, plasma, and/or energy beam loads at various external conditions, such as strong/week magnetic field; physics of Z-pinch, dense plasma focus, and Theta-pinch machines; physics of inertial fusion, and chamber cleaning technologies. Being self-consistent, HEIGHTS includes built in library for all required material, atomic, and plasma property data needed for the calculations.

References:

(There are over 70 publications regarding various capabilities of HEIGHTS): Sample:

(Magneto-)Hydrodynamics, plasma/conducting liquid flow

1.I. Konkashbaev and A. Hassanein, MHD problems in free liquid surfaces as plasma-facing materials in magnetically confined reactors, Fus. Eng. & Des. **61-62**, pp. 223-229 (2002).

2.A. Hassanein, V. Sizyuk, V. Tolkach, V. Morozov, T. Sizyuk, and B. Rice Simulation and Optimization of DPP Hydrodynamics and Radiation Transport for EUV Lithography Devices, Proc. of SPIE Vol. **5374**, 413-422 (2004) .

3.V. Morozov, V. Sizyuk, A. Hassanein, and V. Tolkach, Simulation of Discharge Produced Plasma and EUV Radiation in Various Z-pinch Devices, Argonne National Laboratory, Report ANL-ET-04/31 (2004).

Radiation Transfer

1.A. Hassanein, et al, HEIGHTS Initial Simulation of Discharge Produced Plasma Hydrodynamics and Radiation Transport for Extreme Ultraviolet Lithography, J. of Microlithography, Microfabrication, and Microsystems, **3**(1), 130-138 (2004).

2.A. Hassanein and I. Konkashbaev, Two Dimensional Radiation Transport Method Basing on Multiray Escaping Factor, Computational Physics 2002 and 2002 Annual Meeting of the APS Division of Computational Physics, San Diego, CA, 25-28 August 2002.

Atomic and Plasma Properties; Equation of State

1.V. Tolkach, V. Morozov, and A. Hassanein, Development of Comprehensive Models for Opacities and Radiation Transport for IFE Systems, Argonne National Laboratory, Report ANL-ET/02-23, 2002.

2.V. Morozov, V. Tolkach, and A. Hassanein, Calculation of Tin Atomic Data and Plasma Properties, Argonne National Laboratory, Report ANL-ET-04/24, 2004.

Component Lifetime, Thermal Response, and Erosion Rates

1.A. Hassanein, et al, Nucl. Instrum. And Meth, **B13**, 25 (1986).
2.A. Hassanein and I. Konkashbaev, Theory and Models of Material Erosion and Lifetime during Plasma Instabilities in a Tokamak Environment, Fus. Eng. and Des. **51-52**, p. 681 (2000).
3.A. Hassanein, I. Konkashbaev, and L. Nikandrov, Heat and particle fluxes from collisionless scrape-off-layer during tokamak plasma disruption, J. Nucl. Mater. **290-293**, pp.1079-1083 (2001).
4.A. Hassanein and V. Morozov, Development of Comprehensive and Integrated Models for Inertial Fusion Cavity Dynamics, Argonne National Laboratory, Report ANL-ET/02-04, February, 48, 2002.

Specific issues

1.I. Konkashbaev, A. Hassanein, Ju. Grebenshikov, and D. Ehst, Tritium inventory and safety issues of the dust and debris generated from plasma-facing materials, 4th Int. Symp. Fus. Nucl. Technology, Tokyo, Japan, April 6-11, 1997.

Physical problem being simulated:

Plasma (magneto-, hydro-) dynamics, plasma flow, heat and radiation transport, plasma-material interaction, lifetime of plasma-facing components, and related issues.

Space and time domain:

Depending upon the problem, solution can be composed from the interchangeable modules, varying by the completeness and complexity. HEIGHTS includes 1D-3D MHD modules in planar or cylindrical geometry, 2D-3D radiation transport in planar, cylindrical, spherical and general geometries; slab, planar, or toroidal targets for plasma-material interaction and erosion mechanisms modeling. May include steady-state and/or time-dependent methods for resolution MHD, diffusion, Collisional-Radiation Equilibrium approximation, heat transfer equations.

Underlying physics equations and processes included:

MHD equations, fluid equations, Maxwell's equations, diffusion equation, equation of radiation transfer, Fokker-Planck kinetic description, Hartree-Fock and Hartree-Fock-Slater atomic methods, CRE approximation with/without splitting atomic levels, Planck/Rosseland averaging of optical coefficients, Coulomb collisions, neutral collisions, material destruction, interatomic potentials.

Numerical models:

A combination of various continuum, kinetic, and particle codes.

Computer science and numerical algorithms:
FORTRAN language is mainly used; visualization is implemented as a module with detailed hardware-dependent routines; a postprocessing module is also available; both explicit and implicit time advance algorithms are implemented in different modules.

Requirements and performance:
Portable package (interactive graphics is also available on most platforms), no specific requirements for the modules, but the state and completeness of the problem lays additional requirements on memory, number of CPUs, and typical runs. Most parts of HEIGHTS can be run in parallel mode. Normally, the time-limiting components are MHD, heat transfer, (magnetic) diffusion, and time-dependent CRE.

Verification and validation:
Benchmarking with know solutions and other codes; validation with experimental data (be very brief and include a few references)

1. Validation and verification of the methods for simplified conditions with known (analytic) solution.
2. For each module, comparison of the results with the results from packages of similar functionality or available measured properties
   a) Cowan average therm HF code, Roothaan-Hartree-Fock) and plasma codes. Extensive comparison is given in reference *V. Morozov, V. Tolkach, and A. Hassanein, Calculation of Tin Atomic Data and Plasma Properties, Argonne National Laboratory, Report ANL-ET-04/24, 2004.*
3. Measurements of the properties and parameters of running devices
   a) K. Bergmann, et al, Highly Repetitive, Extreme-ultraviolet Radiation Source Based on a Gas-discharge Plasma, Appl. Opt., **38** (25), 5413-5417 (1999)
   b) U. Stamm, Gas Discharge and Laser Produced Plasma Sources at XTREME Technologies, International SEMATECH EUV Source Workshop, February 23, 2003, Santa Clara, CA

Interaction with other codes:
Inside HEIGHTS, various modules are interacting using both direct coupling (Monte-Carlo based 3D radiation transport is directly coupled into the 2-3D MHD to simulate the dynamics and output of Discharge Produced Plasma source), HFS atomic method is coupled with the CRE to calculate the plasma properties with splitting atomic levels. The output of coupled MHD-RT can be used for further simulation of the erosion rates of the plasma-facing components. The parts of HEIGHTS can also generate information valuable for other codes: atomic data, plasma properties, equation of state, rate coefficients, opacities and others.

User community:
ANL HEIGHTS team; Limited versions of HEIGHTS are used in Germany (KFA) and Japan (JAERI)

11. BOUT

Abstract:

BOUT (BOUndary Turbulence) was developed primarily by X. Xu at LLNL. The code performs direct simulation of 3D turbulence in tokamak edge plasma by time integration of a system of reduced Braginskii fluid equations in realistic geometry with X-point and divertor regions. The code is used to determine the magnitude and characteristics of the turbulence transport believed to dominate radial transport for edge plasmas. Generation of strong radial convection and formation of plasma blobs is characteristic of the turbulence calculated.

References:

Details of the code are described in X. Xu et al., Phys. Plasmas **7** (2000) 1951. A user manual is available in the code archive. Numerous other papers on applications also exist; see http://www.mfescience.org/ users/xu/bout.html

Physical problem being simulated:

The physical problem simulated is tokamak edge plasma turbulence and transport.

Space and time domain:

The code is 3D in space. The typically used configuration is realistic EFIT-based tokamak edge geometry, but simpler geometries, e.g. cylinder or slab, can be used as well. The code is time dependent, and typical run covers time range of about 0.1 ms, which is sufficient for unstable edge-plasma modes to grow and saturate.

Underlying physics equations and processes included:

The physics is described by reduced Braginskii equations, i.e. fluid equations for a collisional plasma (single-ion-species plus electrons), including particle continuity, momenta, and separate ion and electron temperatures. A simple analytic neutral transport model is included as well. Equations are also solved for the electrostatic potential and parallel magnetic vector potential associated with the plasma fluctuations.

Numerical models:

BOUT uses finite-difference model of Braginskii plasma fluid equations for a strong magnetic field. The auxiliary Poisson-like equation for the vorticity utilizes a Fourier transform in the toroidal direction. Ballooning-coordinates are used to more accurately represent the dominant instabilities in the edge region.

Computer science and numerical algorithms:

BOUT is written almost completely in C, with a small fraction of Fortran.
Post-processing visualization can be done with: (i) Xgrafix package, (ii) by importing data to IDL one can use all IDL visualization/analysis tools, in particular, the IDL data analysis package GKV (developed by W. Nevins at LLNL), (iv) also one can use LLNL-developed program PDBview.
Time integration is implicit – using LLNL-developed Newton-Krylov package PVODE.

BOUT uses finite-difference formulation, with 2nd order of discretization (although some terms use 4th order).

Requirements and performance:
Any parallel machine with MPI and standard C and Fortran compilers, e.g., Linux cluster, Power PC, DEC cluster.
BOUT has a typical memory requirement of 300MB for the executable, and an additional storage needed for a typical problem needs 30MB for grid resolutions of 64x64x64.
A typical run run takes about 10 hours on the NERSC CRAY T3E or LINUX clusters with 64 PEs.
The code is parallelized, and it can be run on a few dozens of CPUs. BOUT has been tested on Sun and DEC workstation clusters, and on the NERSC IBM SP and Cray T3E with two different implementations of the MPI library (MPICH and EPCC).
Most of time is consumed by evaluating the right-hand side of the dynamic equations. More than half it is spent on calculating the ExB drift flow, and on inverting the potential vorticity.

Verification and validation:
A benchmark study with the UEDGE code has been done in an axisymmetric setup.
BOUT results have been compared with experimental spatial and temporal spectra of edge fluctuations, for example, X.Q. Xu et al, Nucl. Fusion **42** (2002) 21.

Interaction with other codes:
Work on direct coupling between BOUT and the edge transport code UEDGE, to explore time evolution on long time scales, is ongoing at LLNL. Some results of that have been demonstrated in the past year (T. D. Rognlien et al. Contr. Plasma. Phys. **44** (2004) 188.

User community:
In the past three years, BOUT has been run by users at LLNL and GA. Also, some analysis of BOUT outputs was performed at the Lodestar Corp. There is an ongoing collaboration with the Culham Laboratory (UK), a graduate student from Culham is learning using BOUT for simulations of the MAST experiment.

12. ABAQUS

Abstract
ABAQUS, heat transfer and stress analysis code, Commercial code, Site license at SNL so present cost is unknown, about $7000/year.

References
User Manual and On-line documentation; see www.abaqus.com

Physical problem being simulated
Heat transfer and stress analysis in PFCs, ABAQUS treats generalized plane-strain cases (ANSYS and other codes do not have this)

Space and time Domain
2D and 3D in space, all geometries, transient and steady state

Underlying physics equations and processes included
General purpose finite element code for engineering analysis, linear and non-linear static or dynamic problems, e.g. structural mechanics and heat transfer

Numerical Models
Numerous specific models, e.g., continuum mechanics with specific treatments for work hardening of materials, or Fourier Law for heat transfer that can also be coupled to mechanics

Computer Science and numerical algorithms
Can include full problem setup and post processing, Sandia Fusion technology uses PATRAN to define geometry and ABAQUS as solver and post processor

Requirements and performance
PC-based version now used; most small 3-D problems runs in minutes; rapid transients can take longer; industry uses ABAQUS for complicated geometry such as full engine blocks

Verification and validation
www.abaqus.com

Interaction with other codes
Various inputs possible; Sandia Fusion Technology uses PATRAN to generate geometry and some user defined codes, e.g., FILM for active solution of heat transfer coefficient at coolant boundary

User Community
A widely used commercial code

13. CATIA

Abstract

CATIA, Commercial 3D CAD package, Dassault Systems, feature-based 3D CAD system used on ITER, Cost--$12K + 2K yearly fee

References

http://www.3ds.com/products-solutions/brands/CATIA/, User manual on line

Physical problem being simulated

geometric model building for use in other codes

Space and time Domain

3D, No time dependence

Underlying physics equations and processes included

None

Numerical Models

None

Computer Science and numerical algorithms

Language unknown, Open GL graphics, no time advance

Requirements and performance

PC, 512 MB, no time limit on this

Verification and validation

None

Interaction with other codes

Output to IGES, translated to SAT file for OPERA Input

User Community

http://www.coe.org/, widely used in aerospace and automotive industries. Used for ITER.

14. PATRAN

Abstract

PATRAN, Commercial CAE code, costs about $4500 maintenance per year

References

www.mscsoftware.com/Products/Products_Detail.cfm?PI=6&S=91#jump

Physical problem being simulated

PFC configuration for heat transfer and stress analysis

Space and time Domain

NA - although solver modules for ABAQUS, ANSYS and others are available

Underlying physics equations and processes included

NA

Numerical Models

NA (CAE graphics)

Computer Science and numerical algorithms

NA

Requirements and performance

NA

Verification and validation

See website

Interaction with other codes

input definition for ABAQUS

User Community

widely used CAE commercial Code

15. ANSYS and FEMLAB

Abstract:
ANSYS-Multiphysics and FEMLAB codes used at UCLA (contact Neil Morley)

ANSYS and FEMLAB are commercial finite element codes for solution of thermal, stress/strain, fluid mechanics and electromagnetic problems in arbitrarily complex-shaped, multimaterial domains.

References:
http://www.ansys.com
http://www.comsol.com/

Physical problem being simulated:
Heat Transfer, Thermal Stress, Deformation, Fluid flow properties (some user-generated MHD capability), Electromagnetics

Space and time domain:
Codes written for 3-D, and can also be executed in 1-D, 2-D. Good solid modeling and meshing tools built in as well as various converters for importing solid models and meshes

Requirements and performance:
Used on PCs and Unix platforms

Interaction with other codes:
Good coupling to various modeling packages.

User community:
Widely used in industry and research.

16. CFD2000

Abstract

CFD 2000 GUI, STORM eqn. solver, Fieldview post-processsor. Suite is known as CFD2000

Sandia has perpetual license and a customized version to allow for free surface heat flux, MPI, &porous media auto-generation of foam structures.$5k/yr maint. Fee

References

User's manual, theory handbook, Storm reference, tutorial book, Fieldview post-processsor manual, website:www.adaptive-research.com

Physical problem being simulated

computational fluid dynamics with conjugate heat transfer, laminar & turbulent flows, volume-of-fluid, chemical reactions, radiation, porous media, Sandia has customized free surface heat flux, MPI and model autogeneration features

Space and time Domain

2 & 3-d space, nanosecond to hours time continuum, cartesian, cylindrical & arbitrary body-fitted coordinates, moving grid capability

Underlying physics equations and processes included

Storm is a Navier-Stokes solver, continuum only, not appropriate for rarified atmospheres

Numerical Models

Predictor-Corrector, PISO equation solver, embedded mesh generator

Computer Science and numerical algorithms

Fortran, Unix-based with HOOPS Windows translation

Requirements and performance

INTEL Pentium4 platform, 1GB RAM, 40 GB hard-disk, new 64-node MPI version available spring 2005 for parallel processing on cluster

Verification and validation

benchmark models with many traditional fluid flow experiments, literature - see website

Interaction with other codes

essentially stand-alone. Can only import IGES geometries, embedded geometry editor & tools. Output to Fieldview and Plot3D. Multiple BC options

User Community

3% share of world CFD market, training classes available, Simunet Corp., perpetual license fee $12k. Used in US and Japan by nuclear power & many industrial enterprises.

17. OPERA

Abstract

OPERA 2D and 3D, Magnetic field and eddy current calculations, Vector Fields Limited, Oxford UK., Commercial code.

References

User Manual that comes with purchase.
http://www.vectorfields.com/files/html/products/opera3d.html

Physical problem being simulated

Transient magnetic fields in fusion devices and associated induced currents and forces

Space and time Domain

2D and 3D in space, slab, cylindrical or toroidal possible, both steady state and time dependent

Underlying physics equations and processes included

Maxwell's equations

Numerical Models

Continuum, electro-magnetic code.

Computer Science and numerical algorithms

Pre and post processors for 3D graphics, finite element code

Requirements and performance

Can run on PC and UNIX. We are using a PC with 4 GB memory and typical problems are solved in 3 hours without parallel processing. The time consuming step is the calculation of vector potential in the mesh (both inside and outside of solid).

Verification and validation

See website

Interaction with other codes

Uses SAT file input (ACIS 3D kernal), output to ABAQUS

User Community

Broad use in other fields.

18. CUBIT

Abstract

CUBIT, Sandia National Laboratories, Meshing code for finite element analysis.

References

CUBIT mesh generation environment. volume 1, users manual, SAND 94-1100 Revise, May 1994 CUBIT mesh generation environment, volume 1 users manual, Sjaardema, Gregory D., et al, SAND94-1100 Revise, Sandia National Laboratories, May 1994, 604

Physical problem being simulated

Meshing of complex solid geometries

Space and time Domain

3D in space, no time

Underlying physics equations and processes included

None

Numercial Models

for finite element creation, variety of polygons

Computer Science and numberical algorithims

Unknown

Requirements and performance

PC and UNIX

Verification and validation

unknown

Interaction with other codes

provides input to finite elements codes

User Community

Sandia Labs and other labs outside of fusion

19. FILM-30

Abstract

FILM-30, Marchall, Theron D. Bechtel Bwxt Idaho, Llc, Idaho Falls, Id thermal hydraulic code for boiling curve

References

"FILM-30: a heat transfer properties code for water coolant", Theron D. Marshall, SAND2001-0629 , Sandia National Laboratories, Albuquerque, NM, FEB 2001, 84 pages

Physical problem being simulated

Calculation of heat transfer coefficients at solid to water boundaries

Space and time Domain

Fixed time solver

Underlying physics equations and processes included

Non-boiling, nucleated boiling and post critical heat flux using correlation for pressure

Numerical Models

Correlations

Computer Science and numerical algorithims

Fortran

Requirements and performance

PC, 5 sec

Verification and validation

Verified against experiments

Interaction with other codes

provides output for ABAQUS

User Community

Sandia, INEEL, CEA (France)

20. PFCHF

Abstract

Flux surface code, Mike Ulrickson, a program to calculate heat flux distribution on PFCs

References

None

Physical problem being simulated

following open field lines outside a plasma to calculate heat flux distribution on PFCs

Space and time Domain

2D, cylindrical, time slice

Underlying physics equations and processes included

Calculate field from vector potential (divergence)

Numerical Models

continuum code, analytic

Computer Science and numerical algorithims

Fortran and MathCAD

Requirements and performance

PC, Unknown memory requirements, calculations take about 5 seconds

Verification and validation

None

Interaction with other codes

Gets input from E-fit or other plasma configuration codes, output to heat flux calculating programs such as Heat 1D

User Community

only Mike Ulrickson

21. HEAT-1D

Abstract

Heat 1D, 1D thermal transport using temperature dependent material properties including radiation loss from surfaces and heat transfer to a heat sink, Mike Ulrickson, has database of material properties.

References
None

Physical problem being simulated
heat transport

Space and time Domain
1D in space, time dependent,

Underlying physics equations and processes included
heat diffusion with boundary conditions

Numerical Models
finite difference solver using Runge-Kutta method, explicit time advance, 1st order in time, 2nd order in space

Computer Science and numerical algorithms
Fortran and Mathcad

Requirements and performance
PC, 1 minute or so to solve typical problems, time step size is limited by space step size

Verification and validation
Benchmarked against finite element codes like Abaqus

Interaction with other codes
input from N dot B code.

User Community
Sandia and Princeton

22. IHC

Abstract

Inverse heat conduction solver, Mike Ulrickson, 1D inverse heat conduction solver with temperature dependent heat properties,

References

Method published, but not code. No user manual

Physical problem being simulated

Calculation of surface heat flux from temperature vs time in a material

Space and time Domain

1D, time dependent

Underlying physics equations and processes included

Greens function solution to inverse heat conduction

Numerical Models

finite difference matrix solver, 2nd order

Computer Science and numerical algorithms

Fortran and MathCAD

Requirements and performance

PC, 5 sec

Verification and validation

Comparison to Heat 1D

Interaction with other codes

Input of surface temperature vs time but can use heat and interior place and calculate heat flux.

User Community

Mike Ulrickson

23. NdotB

Abstract
N dot B, Mike Ulrickson, Vector product given fields and surfaces

References
None

Physical problem being simulated
Mapping of plasma heat flux onto real physical surface

Space and time Domain
3D, no time dependence, toroidal

Underlying physics equations and processes included
N dot B

Numerical Models
analytic

Computer Science and numerical algorithms
spreadsheet (excel)

Requirements and performance
PC

Verification and validation
heat flux profiles agree with experiment

Interaction with other codes
input from flux surface code and output to Heat 1D

User Community
only Mike Ulrickson

24. TMAP7

Abstract:
TMAP7, Glen R. Longhurst

Analyzes tritium (or other gas) inventories in systems, including diffusion, trapping, surface processes, heat transfer, fluid flow, chemical reactions, and radioactive decay. Upgrade of TMAP4.

References:
G. R. Longhurst, October 2004, *TMAP7 Users Manual*, INEEL/EXT-04-02352, Idaho National Laboratory, Idaho Falls, Idaho.

J. A. Ambrosek and G. R. Longhurst, October 2004, *Verification and Validation of TMAP7*, INEEL/EXT-04-01657, Idaho National Laboratory, Idaho Falls, Idaho.

Both of these available as PDF files upon request to code author (gxl@inel.gov)

G. R. Longhurst and J. A. Ambrosek, September 2004, *Verification and Validation of TMAP7*, 7th International Conference on Tritium Science and Technology, September 12 – 17, 2004, Baden Baden, Germany.

Physical problem being simulated:
The TMAP Code was written at the Idaho National Engineering and Environmental Laboratory in the late 1980s as a tool for safety analysis of systems involving tritium. It is a completely general one-dimensional mechanistic code for transport of mass and heat through structures and to and from enclosures that requires all material properties and processes to be entered by the user in an input file. Diffusion boundary conditions may be dissociation-recombination limited, solution-law dependent, fixed concentration, non-flow, or surface-energy dependent. Its most recent upgrade to TMAP7 includes up to three separate traps, up to 10 diffusing species, and up to 30 surface-only species, which may be important in surface molecule formation but do not, themselves, diffuse in the material. Under solution-law dependent diffusion boundary conditions, such as Sieverts' law, TMAP7 automatically generates heteronuclear molecular partial pressures for functional enclosures when solubilities and partial pressures of the homonuclear molecular species are provided. In the new surface-energy dependent diffusive boundary condition mode, TMAP7 allows the user to include a surface binding energy and an adsorption barrier energy explicitly. It includes asymmetrical diffusion between the surface sites and regular diffusion sites in the bulk. All of the previous features for heat transfer, flows between enclosures, and chemical reactions within the enclosures have been retained. One additional feature unique to TMAP7 is radioactive decay for both trapped and mobile species.

Space and time domain:
Diffusion and heat transfer are in 1-D slabs with composite and parallel structures possible. Enclosures surrounding and connecting with structures and fluid flows are 0-D. Flows

Calculational time step is dynamically adjusted for stability and convergence.

Underlying physics equations and processes included:
Fick's law diffusion with Soret effect is used for mass diffusion. Heat transfer is by Fourier conduction within structures and standard convection or gap conduction at surfaces. Chemical reactions use standard or non-standard law of mass action.

Numerical models:
Diffusional/thermal structures have arbitrarily set node spacing.

Computer science and numerical algorithms:
Coding is in FORTRAN; makes use of public license GNU compiler.

Diffusional aspects are solved by inversion of a tridiagonal concentration matrix. Non-linear diffusion boundary and trapping conditions are implemented using Gaussian iteration. Flows to and from, and reactions within enclosures are solved using Newton-Raphson iteration.

Spatial equations use second-order implicit (backward) differencing. Temporal time steps use forward differencing.

A pre-processor posts data from the input file to arrays used in the computational module and writes arbitrary algebraic functions to a callable subroutine. Restart capability is included.

Requirements and performance:
GNU Fortran 77 is a multi-environment program, but TMAP7 was developed particularly for the PC Windows environment. It is used from the Command Prompt (DOS) window and has been demonstrated with Windows XP, Windows ME, and Windows NT. Software has been demonstrated on three different Dell PCs, Optiplex GX260, Dimension XPS R450, and Latitude 600 laptop. GNU Fortran 77 compiler is advertised to work on other platforms including workstations, but was not demonstrated.

No inherent capability for parallel processing.

Verification and validation:
J. A. Ambrosek and G. R. Longhurst, October 2004, *Verification and Validation of TMAP7*, INEEL/EXT-04-01657, Idaho National Laboratory, Idaho Falls, Idaho.

Wide assortment of theoretical problems with known solutions were demonstrated. A number of actual experiments were also successfully replicated.

Interaction with other codes:
TMAP7 is a stand-alone code. An extract utility for generating number pair sequences for plotting calculated results is provided.

User community:
Code is new. So far used by Tony Haasz, University of Toronto Institute for Aerospace Studies, and Rion Causey, Sandia National Laboratory. Predecessor, TMAP4, used more widely.

25. MOLDYN

Abstract:
MolDyn

D.A. Alman modified code from Karsten Albe (Univ. of Illinois) who modified the original by K. Beardmore at Loughborough Univ.

MolDyn is a molecular dynamic simulation code that calculates the evolution of a system of particles under particle bombardment. Validated interaction potentials are used to determine the forces from each particle onto the others and the classical equations of motion are solved for each small (~0.1 fs) timestep to track the results of the interactions as a function of time.

References:
D.A. Alman and D.N. Ruzic, Journal of Nuclear Materials **313-316** (2003) 182-186.
D.A. Alman and D.N. Ruzic, Physica Scripta **T111** (2004) 145-151.
Potentials from:
Brenner, D.W., Phys. Rev. B **42** (1990) p9458.
Canales, M., L.E. Gonzalez, and J.A. Padro, Phys. Rev. E, **50**(5) (1994.) p3656-3669.

No user manual is currently available.

Physical problem being simulated:
Particle bombardment of surfaces at low energies, particularly with hydrocarbon molecules and lithium (both as target and bombarding species), to estimate sputtering and reflection properties of materials.

Space and time domain:
3-D coordinate and velocity space.
Time dependent code modeling from 0.1 fs to ps (depending on desired information).

Underlying physics equations and processes included:
Newton's equations of motion using force balance of many-body interaction potentials including thermal effects. Potentials used: Brenner potential for hydrocarbon and lithium potential from neutral pseudoatom (NPA) potential.

Numerical models:
Molecular dynamics

Computer science and numerical algorithms:
MolDyn is programmed in FORTRAN.
All visualization from postprocessing using auxiliary software.
Explicit time advance.

Requirements and performance:
Windows, Linux, Solaris so far.
Memory required: ~2.5MB
Simulation duration: 0.5 to 12 hours per flight depending on lattice parameters.
Currently runs on a distributed computing network, but has been run in parallel using message passing interface (MPI) on both a Windows network and a Linux cluster.
The time-limiting step is the calculation of forces between pairs of atoms.

Verification and validation:
Reflection coefficient of atomic carbon was compared with VFTRIM and hydrocarbon sticking/reflection data from cavity experiments in:
D.A. Alman and D.N. Ruzic, Journal of Nuclear Materials **313-316** (2003) 182-186.
D.A. Alman and D.N. Ruzic, Physica Scripta **T111** (2004) 145-151.

Interaction with other codes:
Output is typically used directly, but additional output files are generated to be used in MDTRIM.

User community:
Darren Alman and Huatan Qiu from Prof. David Ruzic's Plasma-Material Interaction at the University of Illinois.

26. MDCASK

Abstract:
Name: MDCASK, multiple authors, among them: T. Diaz de la Rubia, M.J. Caturla, T. Price, B. Sadigh, G. Gilmer, E. Bringa, M. Duchaineu (LLNL), S. Zybin (CalTech), B. Wirth (UC Berkeley).

Molecular dynamics code, multiple problems simulated: radiation damage in structural materials, chemical sputtering, surface evolution, mechanical properties of nanocrystals, dislocation evolution, laser ablation, shocks in glasses and metals, etc.

References:
User manual and tar files available at:
http://www.llnl.gov/asci/purple/benchmarks/limited/mdcask/

Physical problem being simulated:
For example, edge plasma transport, plasma turbulence, sheath formation, plasma chemistry, sputtering, melting, vapor shielding, conducting liquid flow, etc.

Molecular dynamics code, multiple problems simulated: radiation damage in structural materials, chemical sputtering, surface evolution, mechanical properties of nanocrystals, dislocation evolution, laser ablation, shocks in glasses and metals, etc.

Space and time domain:
3D, any desired configuration can be simulated, enclosed inside a tetragonal box, with a number of boundary conditions (periodic, free, shear, etc.).
MD code (particle code), can simulate up to μs time scales for small systems. Typical simulation times for multimillion atoms simulations are tens of ps.

Underlying physics equations and processes included:
A molecular dynamics (MD) simulation consists of the integration of the Newton equations of motion (F=ma) for an ensemble of particles, given an inter-atomic potential between those particles which gives the force F. Potential energy functions are read from tables. The code can be adapted to treat any short range potential, but currently does not have any capability to treat long range (Coulomb) forces. Currently available potentials: Pair potentials, EAM, MEAM, Garofalini, SW, Tersoff, Brenner, AIREBO.

Numerical models:
Molecular dynamics. MDCASK uses the link cell method to efficiently look for neighbors of each of the atoms in the simulation. This domain decomposition is based on the link cell method. Each node has a fixed number of link cells during the simulation. Each node also has a copy of the first layer of link cells of the neighboring nodes. These 'ghost' link cells have to be updated in order to properly calculate the forces on each of the atoms in the simulation. Some communication also exists when an atom changes position such that it has to be moved to a different node. We implemented a novel

combination of (linked cells + linked lists) that made our code faster by a factor of two at the expense of much larger memory usage, but keeping the memory requirements within limits accessible at most computers.

Computer science and numerical algorithms:
MDCASK is written in F77 with MPI+OMP, and uses a fourth order predictor-corrector algorithm for integration.
The output of this code is the atom positions, kinetic and potential energies and atomic stress, plus the atom id and the atom number. In addition re-start files are also written periodically, with double precision. This means that a combination on runtime and post-processing is typically performed.

Requirements and performance:
MDCASK has been used in IBM SP, Alpha clusters, Linux clusters and Silicon Graphics environments.
The memory used by the program depends on the system size. An upper limit would be approximately 1 Gb/processor. A typical single run to obtain chemical sputtering (using ~10,000 atoms and 20,000 steps) would take ~50 CPU hours, but in order to obtain good statistics in the erosion yields thousands of runs are needed. Usually configurations are stored every 500 steps, and for a 50,000 step run, we would require around 6 GB/run. For 300 runs we would need 2 TB of storage.

This code scales linearly with the number of nodes when the computational box is sufficiently large. Linked cell approach leads to communications of one node with all neighboring nodes (in a cubic grid) at every step. Each node will pass passing about 20 doubles per atom located in a particular node. A node will typically pass up to several thousand atoms per step, and a single time step will take from a small fraction of a ps to several seconds. Even for relatively extreme conditions, inter-node messages have not had large influence in the performance of our code.

Our code has been used with up to 2048 CPU's at NERSC, and it has also been used with up to 1980 CPU's for test runs in MCR, and up to 4000 CPUs at Thunder, two new linux clusters at LLNL. In all cases MDCASK showed nearly linear scaling with number of CPU's (exponent ~1.02 gives best fit). A typical memory usage is ~0.1-0.5 GB/CPU. Memory usage also scales linearly with number of CPU's (atoms simulated).

Typically 85-90% of CPU time is spent at the force loop and the rest is due to other subroutines and node-node communication.

Verification and validation:
This code has been used as a benchmark for other codes, and few sample calculations with their results are available with the download. It is very fast, and it is comparable to other more recent codes like LAMMPS [S. Plimpton (SNL)], and PARCAS [B. Averback (UIUC) and K. Nordlund (Finland)].

In MD codes benchmarking is given, among other things, by reproducing energies for different phases and defect types, and it is generally taken for granted in any publication using a given potential.

Few recent references:
1) B.D. Wirth, G.R. Odette, J. Marian, L. Ventelon, J.A. Young and L.A. Zepeda-Ruiz, "Multiscale Modeling of Radiation Damage in Fe-based Alloys in the Fusion Environment", *Journal of Nuclear Materials* **329-333** (2004) 103.
2) "Molecular dynamics simulations of the Hugoniot of single crystal copper". E. M. Bringa *et al.*, Journal of Applied Physics **96**, 3793 (2004).

Interaction with other codes:
A new module has been added that allows calling a parallel Monte Carlo routine, using the same communication scheme which is used for MD. This new tool allows studying the microstructure of alloys in the scale of multimillion atoms.
New modules are continuously being added, without affecting the code scaling. We plan to add few modules to simulate MeV ion bombardment with energy deposition into electronic excitation. They include thermal spike and two-temperature models.
The code has been used to feed information to a number of other codes, including Dislocation Dynamics codes (H. Zbib (WSU), V. Bulatov (LLNL), hydro codes, etc.

User community:
Because the code is freely available on the web, there are a large number of users at numerous universities all over the world. There is no tracking system for downloads, so it is difficult to quote a figure for the number of users. Among users outside LLNL:
1) B.D. Wirth (UC Berkeley)
2) V. Gillete (UNC)
3) M.J. Caturla (Alicante, Spain)
4) H. van Swygenhoven (PSI, Switzerland)

27. MD_ANL

Abstract:
(Code name, authors, and 2-3 sentence summarizing the code and problem(s) simulated) Provides the structure, transport coefficients, sputtering and bubble formation Molecular Dynamics of liquids and solids, developed by Z. Insepov and A. Hassanein

1) Molecular Dynamics (MD) codes that are capable of studying structural and dynamical properties of liquid Li containing He or H, D atoms, at various temperatures and densities.

2) Molecular Dynamics code to calculate the Li, H, He diffusion coefficients. The diffusion coefficients are calculated by obtaining velocity correlation functions and particle displacements.

3) Molecular Dynamics code for calculating the liquid Lithium sputtering yield due to bombardment by slow He ions. This code contains a new model of liquid Li surface that shows the stratification of surface layers.

4) Molecular Dynamics code for calculating sputtering yield from various solid metals due to bombardment by slow H, D, T, and He ions. These codes contain a combination of atomistic code with the continuum mechanics code.

5) Molecular Dynamics code for simulation of gas bubble formation in liquid Li by simulation of cavity growth mechanisms in bulk liquid Li system with 3-D Periodic-Boundary Conditions.

6) Molecular Dynamics codes for simulation of hydrogen cluster formation in liquid Li.

7) Monte Carlo codes for simulation of gas bubble formation in liquid metals and small hydrogen, helium cluster formation in liquid Li.

References:

1) Z. Insepov and A. Hassanein, "Molecular dynamics simulation of Li surface erosion and bubble formation", submitted for publication in Journal of Nuclear Engineering.

2) M. Terasawa, Z. Insepov, T. Sekioka, A. Valuev, T. Mitamura, "Sputtering due to Coulomb explosion in highly charged ion bombardment", Nucl. Instr. Meth. in Phys.Res. B: 212 (Dec. 2003), pp. 436-441.

3) J.P. Allain, A. Hassanein, T. Burtseva, A. Yacout, Z. Insepov, S. Taj, B. Rice "Radiation-induced synergistic effects of athermal and thermal mechanisms on erosion and surface evolution of advanced electrode and condenser optics materials", in "Emerging Lithographic Technologies VIII", ed. By R.S. Mackay, Proc. of SPIE Vol. 5374 (SPIE,Bellingham, WA, 2004), pp. 112-121.

4) Z. Insepov, J.P. Allain, A. Hassanein, M. Terasawa, "Surface erosion by highly-charged ions", submitted to Nucl. Inst. Meth. in Phys. Res.B.

5) J.P. Allain, A. Hassanein, M. Nieto, Z. Insepov et al., "Xe irradiation effects on single and multi-layer EUV Lithography thin-film optical surfaces", submitted to Nucl. Inst. Meth. in Phys. Res.B.

6) Z. Insepov, A. Hassanein, "Molecular dynamics simulation of Li surface erosion and bubble formation", submitted to the Journ. of Nucl. Mat. (JNM)

7) A. Hassanein, J.P. Allain, Z. Insepov, I. Konkashbaev, "Plasma/Liquid-Metal Interactions during Tokamak Operation", submitted to Fusion Sci. and Technology.

Is a user manual available? How does one access it?
These codes were first developed last year and therefore they are still in an early development stage. Therefore, there are no comments and/or manuals and there is no access to the codes.

Physical problem being simulated:
Solid metals (Cu, W, Mo, Si) surfaces sputtering; liquid Lithium sputtering, hydrogen, helium diffusion, bubble formation, hydrogen cluster formation in liquid Lithium.

Space and time domain:
3D in space, rectangular slab, particle dynamics code, Molecular Dynamics, Langevin dynamics

Underlying physics equations and processes included:
Molecular Dynamics; Monte Carlo; MD code linked to a continuum mechanics code.

Computer science and numerical algorithms:
FORTRAN, visualization by POV-ray software, explicit 2nd order time advance

Requirements and performance:
What platforms will the code run on? Any platform that supports Fortran compiler
How much memory is required? 1 GB, most of the memory goes to multiple linked lists
How long does a typical simulation take? -- 1 – 2 days
Can the code run in parallel? If so, how many CPUs can be utilized and what platforms have been used in the parallel mode? -- The codes have been developed for a single processor and parallelization is being planned in the near future.
What is the major time-limiting component in the code? -- Force calculation, as in any other MD code.

Verification and validation:
The code was benchmarked with known codes and thoroughly validated with experimental data

Interaction with other codes:
It is desirable to couple the results of this code to hydrodynamic codes e.g. HIGHTS

User community:
Z. Insepov is the sole user as this code is under development.

28. ITMC

Abstract:
ITMC – Ion Transport in Materials and Compounds by A. Hassanein and V. Morozov. ITMC code is a 3D Monte-Carlo based simulation package, designed to model the interaction of the incoming ion beam with the target. It can be used to generate the sputtering, implanting, energy loss, and other characteristics of ion/material interaction and mixing.

References:
A. Hassanein, et al, Nucl. Instrum. & Meth, B13, 25 (1986)
User manual is available upon request from the authors.

Physical problem being simulated:
Electron and ion energy loss, stopping power, sputtering, implanting, damage production, angular distribution, ion beam mixing

Space and time domain:
3D in space

Underlying physics equations and processes included:
Coulomb collisions, atomic potentials

Numerical models:
Monte Carlo method

Computer science and numerical algorithms:
Fortran, graphical postprocessing

Requirements and performance:
Portable, less than 5MB, typical simulation strongly depends on the input parameters and number of simulations for statistical post processing and vary from ten minutes to several hours. Admits high-efficient parallel implementation provided the random number generator works differently at different processor nodes.

Verification and validation:
Benchmarked and validated by comparison with the available experimental data and similar packages, such as TRIM.

Interaction with other codes:
Can supply the calculated results (sputtering coefficient, implanting ranges and energy loss) as input to the other codes, such as HEIGHTS-IFE.

User community:
ANL, HEIGHTS Team.

29. VFTRIM-3D

Abstract:
VFTRIM-3D (Vectorized Fractal TRIM in 3-D)
M. Shaheen & D.N. Ruzic added surface roughness and vectorization abilities to existing TRIM.SP by W. Eckstein.

VFTRIM uses Monte Carlo techniques and the binary collision approximation to simulate particle bombardment of targets. It is primarily used to calculate physical sputtering yields, reflection coefficients, and the energy and angular distributions of sputtered and reflected particles. VFTRIM has the option of simulating the same for a rough surface.

References:
The discussion of TRIM.SP – the foundation of the code -- can be found in:
Eckstein, W., *Computer Simulation of Ion-Solid Interactions*. Springer Series in Materials Science, ed. U. Gonser, et al. Vol. 10. 1991, Berlin: Springer-Verlag.

Use of fractal algorithm to model surface roughness:
Ruzic, D.N., *The Effects of Surface Roughness Characterized by Fractal Geometry.* Nuclear Instruments and Methods in Physics Research, 1990. **B47**: p. 118-125.

There is no user manual is available to date.

Physical problem being simulated:
Physical sputtering of a static multilayer/multi-component target.

Space and time domain:
Cartesian 3D in both physical and velocity space.
Static target (time-independent).

Underlying physics equations and processes included:
VFTRIM tracks incident particles (treated as neutrals) and subsequent recoil particles created within a collision cascade from particle bombardment of a (possibly multi-layer and multi-component) target at specified energies and angles of incidence. It only considers 2-body reactions using the Kr-C interaction potential, conserving both energy and momentum. To account for multi-body collisions, VFTRIM considers "soft collisions" which do not create recoils, but reduce projectile energy. Both local and non-local electronic energy losses are modeled in addition to particle energy loss via phonons. It tracks incident and recoil particles moving one mean free path between collisions until all particles have energy below some cutoff value. A planar surface potential is also used to approximate the bulk target binding energy. Particles that leave the surface must have energy perpendicular to the surface greater than this surface binding energy to be considered sputtered.

Numerical models:
Both the impact parameter and the azimuthal angle (in CM frame) are determined by Monte Carlo methods. The “magic formula” is used to calculate the scattering angle from the impact parameter without computing the scattering integral.

Computer science and numerical algorithms:
VFTRIM is written in FORTRAN.
It has no visualization/graphics of its own; it instead has various options for output files.

Requirements and performance:
VFTRIM started in the UNIX environment and has recently moved to Windows for convenience. Very little (~ 1MB) memory is required and a typical simulation is around 2 hours for 10^5 flights at a single energy and angle of incidence for a single Pentium 4 machine.

While VFTRIM can be run on up to 64 processors, it is no longer necessary as a standard desktop completes the simulation fast enough.

The time-limiting part of the code is the tracking of all particles in the cascade for each flight.

Verification and validation:
References showing comparison with experimental data include:
Allain, J.P. and D.N. Ruzic, *Measurements and modeling of solid phase lithium sputtering.* Nuclear Fusion, 2002. **42**: p. 202-210.
Ranjan, R., et al., *Absolute sputtering yield of Ti/TiN by Ar^+/N^+ at 400-700 eV.* J. Vac. Sci. Technol. A, 2001. **19**(3): p. 1004-1007.

TRIM.SP has been thoroughly compared to literature data in:
Eckstein, W., et al., *Sputtering Data.* IPP Report 9/82, 1993.

Interaction with other codes:
All of the interactions to date have been indirect through use of output files. In particular, look-up tables are typically generated for impurity-tracking codes.

User community:
VFTRIM is primarily used by the Plasma-Material Interaction research group (under Prof. David Ruzic) at the University of Illinois at Urbana-Champaign.

30. MD-TRIM-3D

Abstract:

MD-TRIM-3D (Molecular Dynamics supported TRIM)
Authors: J.P. Allain, D.N. Ruzic, D.A. Alman, M.D. Coventry, M.N. Nieto, H. Qiu

Built upon VFTRIM by M. Shaheen & D.N. Ruzic added surface roughness and vectorization abilities to existing TRIM.SP by W. Eckstein.

MD-TRIM combines the realistic multi-body interaction of molecular dynamics simulations with the statistical advantages of Monte Carlo techniques. The code reads a table of collision information generated from MD simulation data of recoil atom creation events and substitutes it for the initial collision events in a VFTRIM simulation. It is based on observations of MD simulations that as sample temperature is increased the resulting scattering angle and transferred energy varies.

References:

MDTRIM:
Allain, J.P., et al., *A model for ion-bombardment induced erosion enhancement with target temperature.* Nucl. Instr. Meth. Phys. Res. B, Submitted 2004.
VFTRIM:
Ruzic, D.N., *The Effects of Surface Roughness Characterized by Fractal Geometry.* Nuclear Instruments and Methods in Physics Research, 1990. **B47**: p. 118-125.
TRIM.SP
The discussion of TRIM.SP – the foundation of the code -- can be found in:
Eckstein, W., *Computer Simulation of Ion-Solid Interactions*. Springer Series in Materials Science, ed. U. Gonser, et al. Vol. 10. 1991, Berlin: Springer-Verlag.

There is no user manual is available to date.

Physical problem being simulated:

Physical sputtering of a static multilayer/multi-component target at elevated temperatures. Incorporating multi-body effects into an otherwise 2-body code.

Space and time domain:

Cartesian 3D in both physical and velocity space.
Static target (time-independent).

Underlying physics equations and processes included:

During MD simulations of liquid metal, initial and final velocities of scattered particles and recoils are tabulated during each PKA creation event, regardless of the number of collision partners involved. MDTRIM reads in these values and substitutes them for the PKA generation routine typically used by a BCA code. Both momentum and energy are conserved in MDTRIM.

Numerical models:
MDTRIM uses a look-up table for PKA creation events and Monte Carlo to sample the values of the table. In addition, higher generation knock-on creation events are treated by Monte Carlo determination of impact parameter and azimuthal scattering angle.

Computer science and numerical algorithms:
MD-TRIM is written in FORTRAN.
It has no visualization/graphics of its own; it instead has various options for output files.

Requirements and performance:
After the molecular dynamic simulation data are available, MD-TRIM requires very little (~ 1MB) memory or computing time. A typical simulation is around 2 hours for 10^5 flights at a single energy and angle of incidence for a single Pentium 4 machine.

The time-limiting part of the code itself is the tracking of all particles in the cascade for each flight; but the MD simulations require orders of magnitude more computing time.

Verification and validation:
MDTRIM shows the qualitative behavior of temperature-enhanced sputtering seen in similar experiments but needs low-energy self-sputtering data to be benchmarked properly.

Allain, J.P., et al., *A model for ion-bombardment induced erosion enhancement with target temperature.* Nucl. Instr. Meth. Phys. Res. B, Submitted 2004.

References showing comparison of VFTRIM with experimental data include:
Allain, J.P. and D.N. Ruzic, *Measurements and modeling of solid phase lithium sputtering.* Nuclear Fusion, 2002. **42**: p. 202-210.
Ranjan, R., et al., *Absolute sputtering yield of Ti/TiN by Ar^+/N^+ at 400-700 eV.* J. Vac. Sci. Technol. A, 2001. **19**(3): p. 1004-1007.

TRIM.SP has been thoroughly compared to literature data in:
Eckstein, W., et al., *Sputtering Data.* IPP Report 9/82, 1993.

Interaction with other codes:
MDTRIM has not interacted with other codes to date.

User community:
MD-TRIM is currently only used by the Plasma-Material Interaction research group (under Prof. David Ruzic) at the University of Illinois at Urbana-Champaign.

31. HIMAG

Abstract:

HIMAG (HyPerComp Incompressible MHD solver for Arbitrary Geometry) is free-surface MHD code by Ramakanth Munipalli, Christopher Rowell, Vijaya Shankar, Carlos Chandler (HyPerComp Inc.,) Mingjiu Ni, Neil Morley, Sergey Smolentsev, Manmeet Narula (UCLA).

HIMAG is a parallel, unstructured mesh-based free surface MHD solver intended for use in fusion related applications. It has been used in cases involving high Hartmann numbers both for single phase as well as two-phase flow applications starting from conceptual design studies in the APEX program, and continued use for NSTX, DiMES and other PFC related projects. Features such as conducting walls, and the anisotropic conductivity for plasma current injection, have been incorporated to make HIMAG well suited to PFC needs.

References:

Preliminary overview: Morley N.B, Smolentsev, S., Munipalli, R., Ni. M.-J., Gao, D., Abdou, M.A., "Progress on the modeling of liquid metal free surface MHD flows for fusion liquid walls," Fusion Engineering and Design, Vol. 72, pg. 3-34, 2004
Further and more detailed publications will be prepared during 2005, including a user manual.

Physical problem being simulated:

Conducting liquid metal flow, with possible plasma interactions via anisotropic flow of current, large deformations, conducting walls in complex geometries.

Space and time domain:

The code is written for 3-D, and can also be executed in 1-D, 2-D and fully developed flow modes. There is no restriction on the geometry. The mesh used can be arbitrary (hexahedra, prisms, tetrahedra). Flow time scale is modeled by second order accurate time stepping. The code solves continuum equations only.

Underlying physics equations and processes included:

Navier-Stokes equations with level set technique for free surface capture, with induction equations for the magnetic field as well as inductionless (electric potential) models for low magnetic Reynolds numbers.

Numerical models:

Crank-Nicholson scheme with mesh collocation and the projection strategy for the Navier-Stokes equations. A higher order TVD scheme for the free surface level set equations with mass conservation corrections. Conjugate gradient as well as Gauss-Seidel based techniques to solve Pressure Poisson Equation and the electric potential Poisson equations. Divergence cleaning procedures for current and magnetic field. Strict enforcement of conservation at material interfaces. Multiple solid walls are admissible.

Computer science and numerical algorithms:
The code is written in FORTRAN and C. Pre-processing, preliminary mesh generation tools for rectangular geometries and an import option from commercial software are available. The code is implicit to second order in time and space. It is based on a finite volume scheme. Graphical user interfaces are provided for all stages of code execution.

Requirements and performance:
The code will run on computer clusters with LINUX, IRIX, SOLARIS and other such operating systems. Memory allocation is dynamic and depends upon the size of the problem. Larger simulations of very high Hartmann number can take a few days, while smaller ones can be completed in a few minutes. The code can be run in parallel across an unlimited number of CPUs. MPI has been used in parallelizing the code. The time limitation comes from the large mesh sizes required in MHD problems, and the need to converge multiple elliptic solvers to a fine degree of tolerance.

Verification and validation:
Several benchmark calculations have already been made and there are important ongoing studies in which comparisons are being made against experiments as well as other solvers. Other solvers include fully developed flow solutions (traditional,) and core flow codes developed in FZK (Germany).

Interaction with other codes:
Coupling options with structural analysis codes and particle based plasma codes are being sought and investigated.

User community:
Approximately 6 users have been actively involved with this code in the past 3 years of its development. Development began in 2001. 3 users from HyPerComp Inc., and 3 users from UCLA have been involved with this code. A version with further developed user interface is expected in the later part of 2005 for a wider community of users, including applications in the metallurgy community in casting and other MHD process development.

32. METAFLOW

Abstract:

MetaFlow is a CFD code being developed by MetaHeuristics in Santa Barbara. The equations governing the fluid flow are solved using the lattice Boltzmann approach and there is a strong emphasis on enabling the code to run efficiently on parallel computer architectures.

References:

M.J. Pattison and S. Banerjee (2004) Numerical Simulation of fluids using the lattice Boltzmann scheme. NUTHOS-6 Conference, Nara, Japan, 4—8 October 2004

Best way to find more information is to e-mail me martin.pattison@gte.net

Physical problem being simulated:

Flow of fluids, including situations with complex geometries, heat transport and MHD effects (as may be found with liquid metal flows).

Space and time domain:

3-dimensional in space
Flexible configuration including complex geometry
Steady-state or time-dependent

Underlying physics equations and processes included:

Fluid equations (solved with lattice Boltzmann method)
MHD equations
Additional scalar transport equation for heat transfer or passive scalar dispersion

Numerical models:

Continuum code – lattice Boltzmann solution scheme.
Body fitted grids are not required for this scheme. Can just use regular cubic grid.

Computer science and numerical algorithms:

Coded in C++
Explicit scheme
Visualisation/graphics not yet included

Requirements and performance:

Not platform specific

How much memory is required?

Problem dependent – no lower limit

How long does a typical simulation take?

Problem dependent; can take minutes, hours, days etc.

Can the code run in parallel? If so, how many CPUs can be utilized and what platforms have been used in the parallel mode?

The lattice Boltzmann scheme was chosen for this work because it is well suited to parallel execution. In principle, there is no limit on the number of processors. I have tested it on a Linux cluster with distributed memory with up to 32 CPUs.

What is the major time-limiting component in the code?

Solution of fluid dynamics equations.

Verification and validation:
Tested against analytical solution for Poiseuille flow, flow over a backward step, flow around a sphere, and a few other cases.

B.F Armaly, F. Durst, J.C.F Pereira and B. Schönung (1983) Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.*, **127**, pp. 473-496

M. Gergova (2002) Evaluation *of improved boundary conditions for the lattice Boltzmann approach: Investigation of the laminar vortex sheet behind a circular cylinder*. Bachelor's Thesis, Friedrich-Alexander University, Erlangen-Nuremberg

Interaction with other codes:
Planning to couple to commonly used pre- and post-processes

User community:
Code is still being developed, but we hope to make it available to people in the fusion community around July. Currently discussing applications with Morley and Abdou at UCLA.

33. FDMHD

Abstract:

FDMHD by Sergey Smolentsev (UCLA). FDMHD is an MHD code for solving complex fluid flow problems for incompressible electrically conducting fluids in a strong magnetic field. The model used includes 2-D MHD equations for a fully developed flow, 3-D energy equation, and 3-D tritium transport equation. The code has specially been designed for blanket applications, for channels with a "sandwich" structure of several materials with different physical properties. The code can be used for the Hartmann numbers up to 10^5.

References:

1. S. Smolentsev, N.B. Morley, M. Abdou, *Code development for analysis of MHD pressure drop reduction in a liquid metal blanket using insulation technique based on a fully developed flow model*, Fus. Eng. Des. 73 (2005), 83-93.
2. S. Smolentsev, M. Abdou, N.B.Morley, M. Sawan, S. Malang, C. Wong, *Numerical analysis of MHD flow and heat transfer in a poloidal channel of the DCLL blanket with a SiC$_f$/SiC flow channel insert*, ISFNT-7, Tokyo, Japan, 2005.

Manual is not available

Physical problem being simulated:
Conducting liquid metal flow in a blanket channel, induced magnetic field, heat transfer, tritium transport.

Space and time domain:
The code solves 2-D momentum equation and 2-D induction equation, while the energy and tritium transport equations are solved in 3-D. The configuration is a multi-layer rectangular channel. Finite-volume formulation and Hartmann number sensitive meshes are used. Heat transfer problem can be treated as time-dependent.

Underlying physics equations and processes included:
Fully developed MHD flow equation coupled with induction equation; convective-diffusion transport equation for the temperature and tritium concentration. User-defined turbulence model can be added to treat turbulent MHD flows.

Numerical models:
The MHD equations are approximated implicitly and solved iteratively using TDMA. A special convergence acceleration relaxation-type procedure is used at high Hartmann numbers. Conservative approximations are used to treat properly interface discontinuities.

Computer science and numerical algorithms:
The code is written in FORTRAN. The code is fully implicit. Finite volume formulation is used. Non-uniform meshes are generated with a built-in mesh generator to cluster grid

points within the Hartmann layers and other sub-domains with significant changes in the magnetic field or flow non-uniformities.

Requirements and performance:
The code will run on the Windows platform. Memory allocation is dynamic and depends upon the size of the problem. Typical cross sectional resolution requires 10^2 x 10^2 mesh points. Larger simulations of very high Hartmann number (~10^5) can take a few days, while smaller ones can be completed in a few minutes.

Verification and validation:
The code has been carefully verified against analytical solutions for both conducting and non-conducting MHD flows available in the literature. Even at Hartmann numbers up to 10^4, the discrepancy is within 0.1%.

Interaction with other codes:
Interaction with other codes has not been investigated.

User community:
The present code and its older versions have been intensively used by its author in the last 6 years.

34. TSLAMHD

Abstract:

TSLAMHD (Thin-Shear-Layer Approximation MHD) was developed by Sergey Smolentsev (UCLA). TSLAMHD is an MHD code for solving fluid flow/MHD equations for open-surface flows in a strong magnetic field using 3-D thin-shear-layer approximation. The model used describes basic MHD effects due to the wall curvature and spatial variations of the applied magnetic field.

References:

3. S.Smolentsev, M.Abdou, *Open-surface MHD flow over a curved wall in the 3-D thin-shear-layer approximation*, Appl. Math. Model. 29 (2005), 215-234.
4. S.Smolentsev, M.Abdou, N.Morley, A.Ying, T.Kunugi, *Application of the k-epsilon model to open channel flows in a magnetic field*, Int. J. Eng. Sci. 40 (2001), 693-711.
5. S.Smolentsev, M.Abdou, T.Kunugi, N.Morley, S.Satake, A.Ying, *Modeling of liquid walls in APEX study*, Int. J. Appl. Electromag. Mech. 13 (2001/2001), 373-379.

Manual is not available

Physical problem being simulated:

Conducting fluid flow (liquid metal or molten salt) with a free-surface over a curved wall in a multi-component, non-uniform magnetic field including heat transfer options. The code had intensively been used in the APEX study.

Space and time domain:

The code solves the flow/MHD and heat transfer equations written in the 3-D thin-shear layer approximation. Using orthogonal body-fitted coordinates and height-function method for tracking free surfaces allows for treating free-surface flows over a curved wall. The code can be applied to free-surface flows, when the interface deformations are moderate (without overturning waves or surface breaking).

Underlying physics equations and processes included:

The mathematical formulation includes the momentum equation written in the 3-D thin-shear-layer approximation, continuity equation, induction equation and the energy equation. User-defined turbulence model (*i.e.* k-epsilon MHD turbulence model) can be added to treat turbulent MHD flows.

Numerical models:

The curvilinear flow domain in the physical plane is reduced to the rectangular domain in the computational plane. All equations are approximated implicitly with the finite volume approach on a non-uniform rectangular mesh. The momentum equation is solved with a marching technique, while the induction and energy equations are solve using the ADI method.

Computer science and numerical algorithms:
The code is written in FORTRAN . The code is fully implicit. Finite volume formulation is used. Non-uniform meshes are generated near the solid boundary and at the free surface. The equations are approximated with the second order accuracy in space and first order in time.

Requirements and performance:
The code will run on the Windows platform. The model developed allows for very fast calculations, which do not take more than a few minutes even at high Hartmann numbers (10^3).

Verification and validation:
The code has been carefully verified against numerical calculations from other codes.

Interaction with other codes:
Interaction with other codes has not been investigated.

User community:
The present code, and its older versions, have been intensively used by the author in the last 6 years.

35. JHS

Abstract:
JHS, 2D and 3D code developed by Ming-Jiu Ni.

JHS is a nonuniform structured mesh-based free-surface MHD solver intended for use in fusion related applications. It has been used in cases involving high Hartmann numbers both for single phase as well as two-phase flow applications.

References

1. M.-J. Ni, S. Komori, AIAA Journal, Vol.40, No.7, pp :1464-1467, 2002
2. M.-J. Ni, S. Komori, N. B. Morley, Numerical Heat Transfer, Part B, Vol. 44, No.6, pp:553-574
3. M.-J. Ni, M. Abdou, S. Komori, Numerical Heat Transfer B, Vol. 44, No.6, pp:553-574
4. M.-J. Ni, S. Komori, N.B. Morley, Int J Heat and Mass Transfer, to be appeared
5. X.-Y. Luo, M.-J. Ni, A. Ying, M. A. Abdou, Numerical Heat Transfer, Part B, to be appeared
6. X.-Y. Luo, M.-J. Ni, A. Ying, M. A. Abdou, Fusion Science and Technology, Vol.47, 1187-1191
7. Morley N.B, Smolentsev, S., Munipalli, R., Ni. M.-J., Gao, D., Abdou, M.A., Fusion Engineering and Design, Vol. 72, pg. 3-34, 2004

Physical problem being simulated:
Simulates conducting single- or multi-liquid metal flow in a magnetic field. Provides an accurate numerical simulation of turbulence by directly numerical simulating flow without any imposed turbulent model. Includes phase change with heat and mass transfer, which can be possibly applied for metallurgical industry.

Space and time domain:
The code is written for 3-D, and can also be executed in 1-D, 2-D and fully developed flow modes. The mesh used can be nonuniform, but structured. Flow time scale is modeled by second order accurate time stepping. The code solves continuum equations only.

Underlying physics equations and processes included:
Navier-Stokes equations with level set technique for free surface capture, with induction equations for the magnetic field as well as inductionless (electric potential) models for low magnetic Reynolds numbers.

Numerical models:
Crank-Nicholson scheme with mesh collocation and the projection strategy for the Navier-Stokes equations. A higher order ENO scheme for the free surface level set equations with mass conservation corrections. Multigrid technique as well as ADI based techniques to solve Pressure Poisson Equation and the electric potential Poisson

equations. Divergence cleaning procedures used for current and magnetic field. Strict enforcement of conservation at material interfaces.

Computer science and numerical algorithms:
The code is written in FORTRAN. The code is implicit to second order in time and space. It is based on a finite difference and finite volume scheme.

Requirements and performance:
The code will run on computers with LINUX, WINDOW and other such operating systems. Memory allocation is dynamic and depends upon the size of the problem. Larger simulations of very high Hartmann number can take a few days, while smaller ones can be completed in a few minutes. The code can only be run in serial computer The time limitation comes from the large mesh sizes required in MHD problems, and the need to converge multiple elliptic solvers to a fine degree of tolerance.

Verification and validation:
Several benchmark calculations have already been made and there are important ongoing studies in which comparisons are being made against experiments as well as other solvers.

Interaction with other codes:

User community:
Approximately 3 users have been actively involved with this code in the past 5 years of its development. Development began in 1999. 2 users from UCLA, and 1 user from Kyoto University have been involved with this code.